UNITED STATES PATENT AND TRADEMARK OFFICE

A

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/038,383	12/31/2001	Sushma Shrikant Trivedi	4860.P2692	7758

7590 11/16/2005

James C. Scheller
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
Seventh Floor
12400 Wilshire Boulevard
Los Angeles, CA 90025-1026

EXAMINER	
HUISMAN, DAVID J	
ART UNIT	**PAPER NUMBER**
2183	

DATE MAILED: 11/16/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/038,383	TRIVEDI ET AL.	
	Examiner	Art Unit	
	David J. Huisman	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *02 September 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-87* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,2,6-16,18,20-29,33-43,45,47-56,60-70,72 and 74-87* is/are rejected.

7)☒ Claim(s) *3-5,17,19,30-32,44,46,57-59,71 and 73* is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *31 December 2001* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-87 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed of record in the file: RCE and Amendment as received on 9/2/2005.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Claim Objections

4. Claim 15 is objected to because of the following informalities: In the last line, insert --of-- after "dispatching". Furthermore, for clarity, it is asked that applicant replace "next" with --branch-- in the last line. Appropriate correction is required.

5. Claim 83 is objected to because of the following informalities: In line 2, replace "registers" with --register--. In line 3, replace "buffers' with --buffer--. Appropriate correction is required.

6. Claim 86 is objected to because of the following informalities: In line 2, replace "registers" with --register--. In line 3, replace "buffers' with --buffer--. Appropriate correction is required.

Withdrawn Rejections

7. Applicant, by way of amendment, has overcome the prior art rejections set forth in the previous Office Action for claims 1-87. Consequently, these rejections are hereby withdrawn by the examiner. However, upon further consideration, a new ground(s) of rejection is applied below.

Claim Rejections - 35 USC § 102

8. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

9. Claims 1-2, 6-8, 15, 21-26, 28-29, 33-35, 42, 48-53, 55-56, 60-62, 69, 75-80, and 82-86 are rejected under 35 U.S.C. 102(b) as being anticipated by Hopkins et al., U.S. Patent No. 5,699,536 (herein referred to as Hopkins).

10. Referring to claims 1, 28, and 55, Hopkins has taught a method for dispatching instructions executed by at least one functional unit of a data processor, each one of the instructions having a corresponding priority number, in a computer system having at least one host processor and host memory, the method comprising:

a) in response to a next instruction having a priority number which is pre-assigned prior to being associated with an instruction group, examining a current instruction group, with respect to the priority number of the next instruction, to determine if the current instruction group is completed, each instruction in the current instruction group associated with a priority number corresponding

to a type of functional unit for executing the respective instruction, and wherein the current instruction group is formed based on a priority number of each instruction in the current instruction group and availability of the corresponding functional unit for executing each instruction. See Fig.8 and Fig.10A. Note that instruction c is of lower priority than instructions a and b because instruction c is dependent on instructions a and b. Therefore, it has been pre-assigned a lower priority than instructions a and b (it is to execute at a later time than a and b). It should also be realized that functional units also play a role in determining priority. For instance, note that instructions a, b, and e had high priority (have no dependencies on each other). As a result, they are all assigned to a functional unit in the same group. However, there are only four (FU0-FU3) functional units. Even if five instructions were to have no dependencies on any instructions, and could therefore be put in a single group, only four would be able to go in a single group, while the 5th will have to wait until a next cycle (lower priority).

b) adding the next instruction to the current instruction group if the current instruction group is not completed based on one or more predetermined conditions associated with the current instruction group. See Fig.10A, and note that each group may hold up to 6 instructions (FU0-BP1). If the group is not completed and a next instruction is not dependent on any of the instructions already in the group, then it may be added to the group.

c) dispatching the current instruction group if the current instruction group is completed, wherein the current instruction group is completed when the one or more predetermined conditions are satisfied, wherein the next instruction is not added to the current instruction group if a corresponding functional unit associated with the priority number of the next instruction is not available for execution. When a group is complete, then that group will issue as is, so that each

instruction in that group will execute. A next instruction is not added to the group if a functional unit is not available. For instance, from Fig.10A, it can be seen that there are a finite amount of functional units. If there are no more available functional units in a given cycle, then even if an instruction is able to be executed that cycle, it must wait since there is no functional unit available.

11. Referring to claims 2, 29, and 56, Hopkins has taught the method of claim 1, wherein if the current instruction group is completed, the method further comprises starting a new instruction group and adding the next instruction to the new instruction group. The examiner asserts that this is inherent. Looking at Fig.10A, it can be seen that independent instructions are added to a first group first. Dependent instructions follow that group. So, if enough independent instructions exist that together use up all the functional units, then there is no choice but to start a new group.

12. Referring to claims 6, 33, and 60, Hopkins has taught the method of claim 1. Hopkins has further taught:

a) examining the current instruction group to determine if the current instruction group contains a predetermined number of instructions. From Fig.10A, it can be seen that each group may hold up to 6 instructions (FU0-BP1). Once there are 6 instructions in a group, no more instructions may be added because there are not enough functional units to accommodate that many instructions per cycle.

b) dispatching the current instruction group if the current instruction group contains the predetermined number of instructions. Once instructions are no longer put in that group, it is free to be dispatched.

13. Referring to claims 7, 34, and 61, Hopkins has taught the method of claim 1. Hopkins has further taught that all instructions in the current instruction group are dispatched in the same clock cycle. See column 8, lines 6-17.

14. Referring to claims 8, 35, and 62, Hopkins has taught the method of claim 1, further comprising:

a) examining the next instruction to determine latency required by the next instruction, wherein the latency is determined based on a type of a functional unit executing the next instruction. From Fig.8-10A, it can be seen that each instruction has a one-cycle delay. More specifically, each functional unit will execute a different instruction every cycle.

b) calculating delay cycles based on the latency. See column 12, lines 21-40, and note that one path may finish execution sooner than the other. Again, it is determined that there is one delay cycle per instruction.

c) suspending the dispatching for a period of time corresponding to the delay cycles. Each group must wait one cycle before it is dispatched. For instance, group 1 will have to wait one cycle for the previous group to finish, group 2 will have to wait another cycle from group 1 to finish.

15. Referring to claims 15, 42, and 69, Hopkins has taught the method of claim 1, wherein if the next instruction is a branch instruction, the method further comprises:

a) examining resources required by the branch instruction to determine if the resources are used or altered by a non-branch instruction. See Fig.8 and note that it is determined that bcx and bcy use some resources being used by instructions a and e, respectively.

b) wherein if the resources are used or altered by a non-branch instruction, suspending the dispatching the next instruction until the resources are available. See Fig.10A, and note that bcx and bcy must be suspended until a and e finish.

16. Referring to claims 21, 48, and 75, Hopkins has taught the method of claim 1, wherein the data processor is integrated in a system core logic chip that functions as a bridge between the host processor and the host memory, and other components of the computer system, the system core logic chip having a host interface coupled to the host processor and a memory interface coupled to the host memory. See Fig.1.

17. Referring to claims 22, 49, and 76, Hopkins has taught the method of claim 1, wherein the data processor may be a stand-alone processor, or the data processor may be a co-processor to the host processor. See Fig.1. There is no mention of this system being a coprocessor so it qualifies as a stand-alone processor.

18. Referring to claims 23, 50, and 77, Hopkins has taught the method of claim 1, wherein the at least one functional unit comprises multiple functional units of a kind. See column 6, lines 28-34.

19. Referring to claims 24, 51, and 78, Hopkins has taught the method of claim 23, further comprising:

a) examining the next instruction to determine if there is a corresponding functional unit that executes the next instruction available. See Fig.10A.

b) adding the next instruction to the current instruction group if the corresponding functional unit is available. See Fig.10A.

c) dispatching the current instruction group without adding the next instruction if the corresponding functional unit is not available. See Fig.10A.

20. Referring to claims 25, 52, and 79, Hopkins has taught the method of claim 24, wherein if the corresponding functional unit that executes the next instruction is available, the method further comprises:

a) examining the next instruction to determine if the next instruction is required to be in a new instruction group, wherein the next instruction is required to be in the new instruction group if a resource used by the next instruction is to be used by at least one instruction of the current instruction group or the next instruction is in a category different than at least one instruction of the current instruction group. See Fig.8-10A, and note that dependent instructions may not be in the same group as the instructions that they are dependent on.

b) wherein if the next instruction is required to be in a new instruction group:

b1) adding a no-operation (NOOP) instruction to the current instruction group to complete the current instruction group. See Fig.10A and note that the blank slots indicate NOPs as no instruction is executed in that specific slot in that cycle.

b2) dispatching the current instruction group without adding the next instruction. Groups are inherently dispatched.

b3) starting a new instruction group. See Fig.10A.

b4) adding the next instruction to the new instruction group. See Fig.10A.

21. Referring to claims 26, 53, and 80, Hopkins has taught the method of claim 24, wherein if the corresponding functional unit that executes the next instruction is available, the method further comprises:

a) examining the current instruction group to determine if the current instruction group contains a predetermined number of instructions. See Fig.10A and note that only 6 instructions per group may exist.

b) wherein if the current instruction group contains the predetermined number of instructions:

b1) dispatching the current instruction group without adding the next instruction. Groups are inherently dispatched.

b2) starting a new instruction group. If there are already 6 instructions in a group, a new group must be started.

b3) adding the next instruction to the new instruction group. Again, if there are already 6 instructions in group A, then a next instruction must go to a group other than group A.

22. Referring to claim 82, Hopkins has taught an apparatus for dispatching instructions executed by at least one functional unit of a data processor, the apparatus comprising:

a) an instruction cache memory for receiving instructions from an input and output (1/0) interface, each instruction associated with a priority corresponding to a type of functional unit for executing the respective instruction, wherein the priority of each instruction is pre-assigned prior to being associated with an instruction group. See Fig.8 and Fig.10A. Note that instruction c is of lower priority than instructions a and b because instruction c is dependent on instructions a and b. Therefore, it has been pre-assigned a lower priority than instructions a and b (it is to execute at a later time than a and b). It should also be realized that functional units also play a role in determining priority. For instance, note that instructions a, b, and e had high priority (have no dependencies on each other). As a result, they are all assigned to a functional unit in the same group. However, there are only four (FU0-FU3) functional units. Even if five

instructions were to have no dependencies on any instructions, and could therefore be put in a single group, only four would be able to go in a single group, while the 5th will have to wait until a next cycle (lower priority).

b) an instruction decoder coupled to construct an instruction group based on the priorities of the instructions and availability of the corresponding functional units, wherein an instruction is not added to the instruction group if a corresponding functional unit associated with the priority number of the instruction is not available for execution when the instruction group is dispatched. See Fig.10A. Note that only 6 instructions may exist in a group. If any more are eligible to execute in a group already having 6 instructions, then that instruction must wait..

c) a dispatch controller coupled to dispatch the instruction group to an appropriate functional unit if the instruction group is completed, wherein the instruction group is completed when one or more predetermined conditions are satisfied. When a group is complete, then that group will issue as is, so that each instruction in that group will execute. A next instruction is not added to the group if a functional unit is not available. For instance, from Fig.10A, it can be seen that there are a finite amount of functional units. If there are no more available functional units in a given cycle, then even if an instruction is able to be executed that cycle, it must wait since there is no functional unit available.

23. Referring to claim 83, Hopkins has taught the apparatus of claim 82, further comprising:

a) at least one instruction registers coupled to store the instructions being grouped. See Fig.10A.

b) at least one instruction buffers coupled to store instructions when the instruction fetching is stalled. See Fig.1, component 111.

24. Referring to claim 84, Hopkins has taught the apparatus of claim 82, further comprising a branch decoder coupled to detect a branch condition and to generate the address for the next instruction being fetched. This is deemed inherent in any system that has branch instructions, which Hopkins has.

25. Referring to claim 85, Hopkins has taught the apparatus of claim 84, further comprising a program counter coupled to receive commands from the branch decoder to fetch the next instruction at the address. A program counter (PC) is an inherent component in any system which points to a next instruction to fetch.

26. Referring to claim 86, Hopkins has taught the apparatus of claim 83, wherein the instruction decoder retrieves the instructions from the at least one instruction registers or from the at least one instruction buffers after the instruction stalling cycles. Clearly, after the machine stalls, more instructions may be received and processed.

Claim Rejections - 35 USC § 103

27. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

28. Claims 9, 11-14, 16, 18, 20, 27, 36, 38-41, 43, 45, 47, 54, 63, 65-68, 70, 72, 74, 81, and 87 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hopkins, as applied above.

29. Referring to claims 9, 36, and 63, Hopkins has taught the method of claim 8. Hopkins has not explicitly taught inserting an additional delay cycle during the suspension. However,

Official Notice is taken that stalling is well known in the art. That is while a next group is suspended (waiting to issue until the previous group is finished executing), the currently executing group may stall due to a hazard. Stalling therefore affects all subsequent groups and is a way to correct the hazards. As a result, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hopkins to insert additional delay cycles during the suspension.

30. Referring to claims 11, 38, and 65, Hopkins has taught the method of claim 1. Hopkins has taught dependencies between instructions (Fig.8) but not specifically that if the next instruction is a non-branch instruction, the method further comprises:

a) examining the next instruction to determine if source resources required by the next instruction are in-use. However, RAW hazards are known in the art to be those in which source registers of a dependent instruction are being used by a previous instruction. This is an extremely common type of hazard, and therefore it would have been obvious to examine instructions for this type of error in Hopkins.

b) stalling instruction dispatching if the source resources required by the next instruction are in-use. It is inherent that stalling must be performed to correct RAW hazards.

31. Referring to claims 12, 39, and 66, Hopkins has taught the method of claim 11, wherein the source resources are defined by source operand registers required by the next instruction. Again, a RAW hazard is known to occur when a dependent instruction's source(s) is/are the destination(s) of previous instruction(s).

32. Referring to claims 13, 40, and 67, Hopkins has taught the method of claim 1. Hopkins has taught dependencies between instructions (Fig.8) but has not specifically taught that if the next instruction is a non-branch instruction, the method further comprises:

a) examining the next instruction to determine if destination resources required by the next instruction are in-use. However, WAW hazards are known in the art to be those in which the destination register of a dependent instruction is being used by a previous instruction. This is a common type of hazard, and therefore it would have been obvious to examine instructions for this type of error in Hopkins.

b) stalling instruction dispatching if the destination resources required by the next instruction are in-use. However, it is known that stalling is a common way to overcome WAW hazards. As a result, it would have been obvious to stall Hopkins so that the WAW is corrected.

33. Referring to claims 14, 41, and 68, Hopkins has taught the method of claim 13, wherein the destination resources are defined by target destination registers required by the next instruction. Again, WAW hazards are known in the art to be those in which the destination register of a dependent instruction is being used by a previous instruction.

34. Referring to claims 16, 43, and 70, Hopkins has taught the method of claim 15. Furthermore, claims 16, 43, and 70 are rejected for the same reasons set forth in the rejections of claims 9, 36, and 63, above.

35. Referring to claims 18, 45, and 72, Hopkins has taught the method of claim 6. Hopkins has further taught that the predetermined number of instructions comprises six instructions (Fig.10A, components FU0-BP1), and not four instructions as claimed by applicant. However, as shown in In re Rose, 105 USPQ 237 (CCPA 1955), changes in size/range are generally not

given patentable weight or would have been an obvious improvement. One of ordinary skill in the art would have recognized that the number of instructions per group is based on the amount of functional units that can handle that group. So, if there are six units, six instructions may execute per cycle (as taught in Hopkins). However, if there are four functional units, only four instructions may execute. The number of functional units is completely based on designer choice. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hopkins to have just four execution units instead of six so that the predetermined number would be four.

36. Referring to claims 20, 47, and 74, Hopkins has taught the method of claim 8. Hopkins has not taught accessing a database to determine the latency required by the next instruction. However, Official Notice is taken that storing latencies to assist in scheduling is well known. And databases, as known, are efficient storage structures. As a result, in order to assist in scheduling, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hopkins to include a database for holding latencies.

37. Referring to claims 27, 54, and 81, Hopkins has taught a method of claim 26. Furthermore, claims 27, 54, and 81 are rejected for the same reasons set forth in the rejections of claims 18, 45, and 72, above.

38. Referring to claim 87, Hopkins has taught the apparatus of claim 82. Hopkins has not explicitly taught that the instruction decoder stalls the instruction fetching based on the latency of the instruction being executed. However, it is well known that when instructions stall the system for some latency, all of the stages prior to that instruction's current stage are stalled as well. If they weren't stalled then data would be lost as some data is trying to progress through the

pipeline while it should be stalled. Consequently, it would have been obvious to one of ordinary skill in the art at the time of the invention to modify Hopkins to stall fetching when an instruction is stalling.

39. Claims 10, 37, and 64 are rejected under 35 U.S.C. 103(a) as being unpatentable over Hopkins, as applied above, in view of Dowling, U.S. Patent No. 6,363,475 (as applied in the previous Office Action).

40. Referring to claims 10, 37, and 64, Hopkins has taught the method of claim 1. Hopkins has not taught, but Dowling has taught:

a) examining the next instruction to determine if the next instruction contains an illegal operation code (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23); and

b) issuing an error message through an interrupt mechanism, if the next instruction contains an illegal operation code (Dowling abstract, figures 4-5, column 3 line 54-column 4 line 6, column 12 line 41-column 13 line 36, column 14 lines 3-23).

Clearly, if an illegal operation exists, and the architecture is unable to execute it, thereby causing the system to fail, then it would have been obvious to modify Hopkins to detect an illegal operation and issue an error message in response to detection.

41. Claims 1-2, 28-29, and 55-56 are rejected under 35 U.S.C. 102(b) as being anticipated by Schepers, U.S. Patent No. 5,712,996 (as applied in the previous Office Action).

42. Referring to claims 1, 28, and 55, Schepers has taught a method for dispatching instructions executed by at least one functional unit of a data processor, each one of the instructions having a corresponding priority number, in a computer system having at least one host processor and host memory, the method comprising:

a) in response to a next instruction having a priority number which is pre-assigned prior to being associated with an instruction group, examining a current instruction group, with respect to the priority number of the next instruction, to determine if the current instruction group is completed, each instruction in the current instruction group associated with a priority number corresponding to a type of functional unit for executing the respective instruction, and wherein the current instruction group is formed based on a priority number of each instruction in the current instruction group and availability of the corresponding functional unit for executing each instruction. See the abstract, Tables 1 and 2 (in column 6, and the process of claim 1. Essentially, instructions are divided into groups based on priority, where priority is determined based on a number of factors, the factors being delay cycles, succeeding instructions, and distance value. See column 4, lines 33-51. In addition, it is inherent that a group must be examined for completion because if it is completed (i.e., 3 instructions have been assigned to it), then a next instruction cannot be assigned to the group. It should be realized that priorities are based on functional units because there is only a finite number of units. If X instructions may issue in a single cycle according to every other factor but only X-1 units exist for execution, then only X-1 instructions may issue. The system would decide which instruction gets left out, (i.e. which one is lower priority).

b) adding the next instruction to the current instruction group if the current instruction group is not completed based on one of more predetermined conditions associated with the current instruction group. See Table 2 and column 6, line 48, to column 7, line 13, as an example of this. Basically, a first instruction (l.s) is assigned to a first group and a next instruction (lw) is subsequently assigned to the first group since the first group is not yet completed (i.e., there are not 3 instructions assigned to it yet).

c) dispatching the current instruction group if the current instruction group is completed, wherein the current instruction group is completed when one or more predetermined conditions are satisfied, wherein the next instruction is not added to the current instruction group if a corresponding functional unit associated with the priority number of the next instruction is not available for execution. See the abstract and Table 2 and note that when instruction groups are completed, i.e., they have 3 instructions assigned to them, then they are dispatched for execution. One predetermined condition would be whether or not the group has 3 instructions assigned to it. And again, if there are not enough functional units, then an instruction may not be added to the group.

43. Referring to claims 2, 29, and 56, Schepers has taught a method as described in claim 1. Schepers has further taught that if the current instruction group is completed, the method further comprises starting a new instruction group and adding the next instruction to the new instruction group. Again, if there is no more room to add an instruction to a group a new group is started and a next instruction is added.

Allowable Subject Matter

44. Claims 3-5, 17, 19, 30-32, 44, 46, 57-59, 71, and 73 are objected to as being dependent upon a rejected base claim, but would be allowable if rewritten in independent form including all of the limitations of the base claim and any intervening claims.

Conclusion

Any inquiry concerning this communication or earlier communications from the examiner should be directed to David J. Huisman whose telephone number is (571) 272-4168. The examiner can normally be reached on Monday-Friday (8:00-4:30).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

DJH
David J. Huisman
October 31, 2005

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100